(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 August 2001 (30.08.2001)

PCT

(10) International Publication Number
WO 01/63385 A1

(51) International Patent Classification⁷: G06F 1/00

(21) International Application Number: PCT/GB01/00688

(22) International Filing Date: 20 February 2001 (20.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0003920.6 21 February 2000 (21.02.2000) GB

(71) Applicant *(for all designated States except US)*: NCIPHER CORPORATION LIMITED [GB/GB]; Jupiter House, Station Road, Cambridge CB1 2JD (GB).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: VAN SOMEREN, Nicko [GB/GB]; nCipher Corporation Limited, Jupiter House, Station Road, Cambridge CB1 2JD (GB).

(74) Agent: HALLAM, Arnold, Vincent; Lewis & Taylor, 144 New Walk, Leicester LE1 7JA (GB).

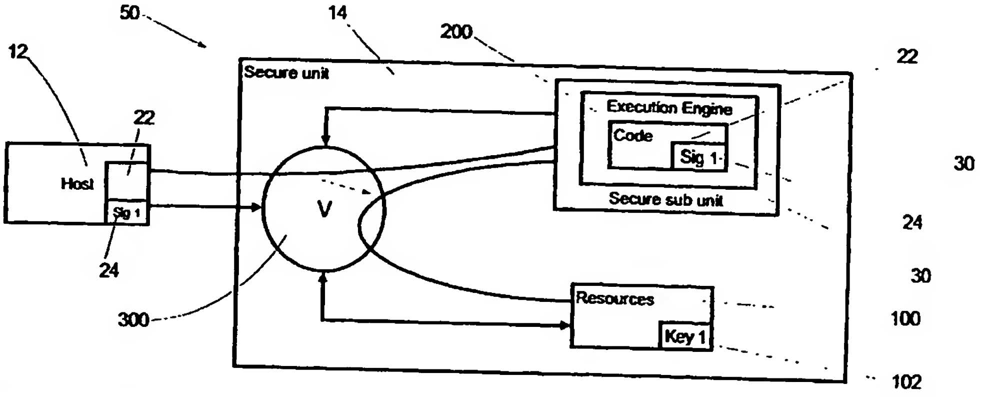
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: CONTROLLING ACCESS TO A RESOURCE BY A PROGRAM USING A DIGITAL SIGNATURE



(57) Abstract: A computer system has a resource, a verification unit and an execution engine for running a body of program code having an associated signature. A cryptographic key is associated with the resource and when the code is to be loaded into the execution engine a verification operation is run on the signature using the cryptographic key associated with the resource. The execution engine is separate from the resource and when access to the resource is required by the code in the execution engine a further verification operation is conducted on the signature using the cryptographic key associated with the resource. Access to the resource by the code depends upon the result of the verification operation.

CONTROLLING ACCESS TO A RESOURCE BY A PROGRAM USING A DIGITAL SIGNATURE

The present invention relates to a computer system and particularly, but not exclusively, to a so-called secure computer system.

In many computer systems, it is necessary to control access to certain resources, such as areas of computer memory, cryptographic keys or other sensitive data. This might be because they contain secret information, because the resource is scarce, because the resource has intrinsic value or a combination of factors. In such systems it may be necessary to allow some parties access to the resources while denying access to other parties.

There are a number of existing systems which are designed to enforce access control to such resources. There are, however, other more demanding systems where simply controlling who can access the resource is not sufficient. In these systems it may also be necessary to control what is done with the resource.

The present invention aims to provide an improved method for securing a resource on a computer system and an improved computer system.

Accordingly, the present invention provides a method of controlling access to a resource in a computer system by a body of code having a signature associated therewith, the method comprising the steps of providing a cryptographic key associated with said resource, conducting a verification operation on said signature using the cryptographic key associated with said resource and controlling access to the resource by the code in dependence upon the result of said verification operation.

The present invention also provides a computer system comprising verification means for verifying a digital signature associated with a body of program code and a resource having a cryptographic key associated therewith wherein said verification means is operable to use said cryptographic key to verify said digital signature thereby to allow access of the code to said resource in dependence upon the verification.

The invention provides a novel system for controlling, on a fine grained basis, what can be done with a resource. In our system the actions will be carried out by a computer program and our invention allows various resources to specify which parties may perform which actions.

The present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a diagrammatic representation of a conventional secure computer system;

Figure 2 is a representation of the operation of the system of Figure 1;

Figure 3 is a representation of a second known computer system;

Figure 4 is a diagrammatic representation of a secure computer system according to the present invention;

Figure 5 is a representation of the operation of the system of Figure 4 ;

Figure 6 is a representation similar to that of Figure 5 of the operation of a modification to the system of Figure 5;

Figure 7 is a representation similar to that of Figure 5 of the operation of a second modification to the system of Figure 5;

Figure 8 is a representation similar to that of Figure 5 of the operation of a third modification to the system of Figure 5; and

Figure 9 is a representation similar to that of Figure 5 of the operation of a fourth modification to the system of Figure 5.

In the drawings like parts are given like reference numbers.

Figure 1 is a representation of a hardware computing system 10 which allows a user access to a sensitive or restricted resource. The system has a host computer 12 connected to a secure module 14 which may be a card in the computer 10 or a separate and remote unit. The secure module 14 contains the resource 100, an execution engine 200 and an access control unit 300. The system 10 may constitute, for example, a main frame or server PC of a business such as a bank which may be accessed internally or by way of an on-line access, e.g. by a customer. The execution engine 200 is code which is running in the secure module 14 and has access to the security sensitive resource 100 which typically may be bank account data stored in Random Access Memory (RAM).

In use, when a third party attempts to gain access to the resource 100 via the host computer 12 the latter applies a computer program code 22 to the execution engine 200 to gain access to the resource 100. The code 22 to be executed has a digital "signature" 24, in the form of a unique code, encoded either within the program code 22 or presented alongside the program code. The signature is indicated as Sig 1 to show that it is specific to a particular individual or group of individuals referred to here as the primary owner. Before the code is loaded into the execution engine 200 the signature is verified in the access control unit 300 using a cryptographic key (Key 1) and only if the verification is positive is the code allowed to load into the execution engine to give the third party access to the resource 100.

Figure 2 is a representation of the operation of the system software and shows the program code 22 and associated digital signature 24 in the host computer 12, and the resource 100, execution engine 200 and access control unit 300 in the secure module 14. The access control unit 300 includes an access control device 302 which controls access to the resource 100.

In use, the program code 22 to be executed by the execution engine 200 is applied to the access control unit 300. As indicated above, the code 22 to be executed has a digital "signature" 24, in the form of a unique code, encoded either within the program or presented alongside the program. The access control unit 300 contains a cryptographic key 304 (Key 1), which may be referred to as the primary key, and a signature verifier 306 in the form of a comparator operable to check the signature 24 associated with the code. The cryptographic key 304 is set by the manufacturer of the secure module 14. The signature verifier 306 checks the signature 24 using the cryptographic key 304 in the access control unit 300. Providing the signature 24 on the code is correct, the access control unit 300 allows the code to be loaded into the execution engine 200 where it can be run and gain access to the resource.

Thus, the verification of the signature relies upon key 304 that is built into the access control unit 300 by the module manufacturer and it is the loading of the code 22 into the execution engine 200 which is controlled by the verification process.

Figure 3 shows a conventional variant of the system of Figure 2 where the primary owner wishes to delegate access to the resource 100 to another party. The code 22 contains a delegation certificate 101 consisting of a further cryptographic key 42 (Key 2) and the signature 24 of the primary owner, possibly accompanied by other data. The code also contains a new signature 26 (Sig 2) for the other party, referred to here as the secondary owner, to whom the primary owner of the first signature 24 wishes to delegate access to the resource. In this system the key 304 (Key 1) built into the access control unit 300 uses the verifier 306 to verify the signature 24 of the primary owner on the certificate 101 while a second verifier 308 checks the second signature 26 of the secondary owner on the code against the key 42 of the primary owner in the certificate 101. The outputs of the two verifiers are taken together using an "and" operation 310 and only if the first signature AND the second signature are verified is the secondary owner allowed access to the resource 100 by the access control device 300.

A common property of these conventional systems is that ultimately the verification of the signature, or the chain of delegation certificates, ends with a key that is built into the system. It is up to the owner of the system which keys are to be trusted for which operations.

Figure 4 is a representation similar to that of Figure 1 of a preferred form of hardware computer system 50 according to the invention. In Figure 4, the secure module 14 contains a secure sub unit 30 in which the execution engine 200 is located. The access control unit 300 and the resource 100 are separated from the execution engine 200 within the module 14 so that the execution engine 200 operates within the secure sub-unit 30. The separation of these devices may be physical, by means of controlling the electrical and mechanical connections between the devices, or through logical separation in the operation of software components within the computer system. The resourcc 100 also contains a cryptographic key 102 (Key 1) which is the primary key and is similar to key 304 of Figures 2 and 3. However, the key 102 is set by the owner of the resource 100 and not by the manufacturer of the module 14. It will be appreciated that, as mentioned above, although 14 is referred to as a module it may be in the form of a card contained within the host computer 12 or may be one or more discrete units physically separate from the host computer 12.

The host computer 12 may communicate with the execution engine 200 only through the access control unit. As shown in Figure 4, any code operating within the secure sub-unit 30 is unable to access any resource outside the secure sub-unit 30 other than by issuing a request through the access control unit 300.

When the host system 10 attempts to load the program code 22, togther with its associated authentication signature 24, into the execution engine 200, this code is first passed through the access control unit 300 which checks that the signature 24 associated with the code 22 is a valid signature using the key 102. At this point, the access control unit 300 does not associate this signature with any authority to access the resource 100. However, the access control unit 300 records the identity of the signature creator and associates this with the program code 22 that has been loaded. When the program code 22 is subsequently executed by the execution engine 200 within the sub-unit 30, the code may request access to the resource 100 and the request is passed through the access control unit 300. The identity recorded against the code at the time that it was loaded is then checked by the access control unit 300 using an access control policy to ascertain what actions, if any, are authorised by the signature 24. Only those authorised actions are allowed to be carried out on the resource 100.

Figure 5 is similar to Figure 2 and is a representation of the operation of the system software within the module 14.

In the system of Figure 5, the resource has associated with it the cryptographic key 102 which may be used to verify the digital signature 24 associated with the body of program code 22. The program code, along with its associated digital signature 24 is first loaded into the module 14 where the access control block 300 compares the signature with the key 102 associated with the resource 100. The access control unit 300 includes access control device 302 which controls access to the resource 100, and signature verifier 306. If the signature is correctly verified then the code is allowed to load in the execution engine 200 but still does not have access to the resource 100 and the access control unit 300 records the identity of the signature creator and associates this with the program code 22 that has been loaded.

If the user then requires access to the resource 100 to carry out a particular operation, the code 22 is run in the execution engine 200 and requests access to the resource 100. This access has to be effected through the access control unit 300 and at this stage the signature 24 is checked against the access control policy.

In its simplest form in Figure 5 the signature 24 is verified against the key 102 by the verifier 306 and if verified access is allowed to the resource 100. Thus, it is the resource itself that determines which code is allowed access to the resource because the key 102 is in the resource and not the module 100. This means that each user can set his own key in his own data block in the resource 100 and this is not therefore accessible by whoever has access to the module 14.

In variants of this invention more than one signature can be applied to the program code using different cryptographic keys. In this case access to the resource will be granted if any of the signatures verify against any of the listed keys given with the resource. The signature on the program code may be replaced by a signature on a cryptographic hash of the program code, a signature on a set of hashes of parts of the program code, on a hash or hashes of the parts of the program code or on any other method which allows the verification of the signature to be completed only if the code is identical to the code upon which the signature was originally placed, since the signature is unique to the code.

The keys associated with the resource may be replaced by cryptographic hashes of those keys or other identifiers of the keys which allow the device uniquely to identify the correct keys with which to verify the signature. In these cases the real signature verification keys must be available to the device in order to carry out the verification process.

In this invention the signature can be any form of digital or electronic signature which can be verified by the device.

The system of Figure 5 does not restrict the actions which the signature holder might carry out on the resource 100 once the code is allowed access to the resource 100 by the access control unit 300, since the key does not have any access restrictions associated with it. Figure 6 shows a system in which the actions are restricted to those allowed by the associated key. In this system, the resource 100 has associated with it an access control list (ACL) 106 comprising key sets 108. Each set 108 indicates some actions which may be performed using the resource, some key which may be used to authorise those actions and, optionally, some constraints upon those actions including time limits, usage limits, parameter limits or other constraints. As before, the resource is made available to the execution engine 200 but this time it is bound to the access control lists rather than simply to keys. The program code 22 is loaded into the execution engine 200 as before along with the signature 24 upon the code. The code may then be executed inside the execution engine 200 and when the code attempts to carry out any operation upon the resource the access control list 106 is checked. If the signature on the code can be verified by the verification unit 306 with respect to one of the keys in the ACL the access control unit 300 allows the code access to the operations listed in the ACL entry which has the correct key, providing the access falls within any given constraints. Otherwise the actions are not permitted. For example, one access key may allow the user read only rights to data in the resource whilst another might allow read/write access. Another possibility where the resource is memory is for different keys to allow access to different parts of memory.

In a third version of the invention (Figure 7) intermediate access control lists may be used to delegate control. The code 22 contains a delegation certificate 400 consisting of a further cryptographic key 402 (Key 3) and the signature 24 of the primary owner. The code also contains a new signature 28 (Sig 2) for the secondary owner, to whom the primary owner of the first signature 24 wishes to delegate access to the resource 100. As in the embodiment of Figure 6 the use of the resource 100, or individual operations on the resource, is associated with signature verification keys 108 of access control list 106. However, in this system delegation credentials may be constructed in the following manner: an access control list 402 in the delegation certificate 400 is built in a similar form to the list 106 associated with the resources. This list 402 is then signed using the cryptographic signature key 24. The resource 100 is made available to the execution engine 200, along with its access control list 106, and the program code 22, its signature 28, and the delegation credential 400, including its signature 24 are all loaded into the execution engine 200. The signatures on the code can be checked by verification unit 308 against the keys in delegation credential 400 and the signature 24 on the delegation credential is verified against the keys 108 in the resource's access control list 106 using verification unit 306. The intersection is taken at 310 of the operations listed in validated ACL entries and validated delegation credentials. Access is granted by access control device 302 to the code for opcrations that fall within this intersection of operations.

It will be appreciated, therefore, that one user with rights to one part of the resource 100 can sign off or delegate part or all of those rights to a third party. Referring again to Figure 7, the output of verifier 308 shows operations identified as [2, 3, 4] being allowed by the verification check of the secondary owner's signaturc 28 against the delegation key 402, for that signature, of the code access control list 400. However, the output of verifier 306 shows only operations identified as [1, 3] being allowed by the verification check of the primary owner's signature 24 against the delegation key 108 in the resource access control list 106 for that signature. Thus, the secondary owner is only allowed to carry out operation [3] on the resource 100.

In a fourth variant of this invention (Figure 8) access can be granted to all code which is authorised under the system shown in Figure 6 as well as code authorised as described above in relation to Figure 7. Multiple levels of delegation may be permitted in which the operation must be listed in each ACL with a key which verifies the signature on the next credential, or which verifies the program code in the last credential.

As shown in Figure 8 this scheme can be used to allow multiple bodies of code 22a, 22b to have differing levels of access to the same resource 100 by having different delegation certificates 400a, 400b and signatures 24a, 24b, 28, 32.

The codes 22a, 22b are loaded into different execution engines 200a, 200b within the secure module 14. It will be appreciated that the different execution engines 200a, 200b, as well as individual resources, may be in different secure sub units within the secure module 14 and cannot therefore interfere with one another.

In a further modification (Figure 9) the system of Figure 8 can be used to allow one body of code 22 different levels of access to more than one resource 100a, 100b.

The code 22 is similar to code 22a or 22b of Figure 8 but has two or more delegation certificates 400a, 400b, each with a cryptographic key 402a, 402b and signature 24, 26. Here there are two primary owners of signatures 24, 26 who both wish to delegate to the same user or secondary owner of signature 28. The secondary user is granted different rights to the two resources 100a, 100b in dependence on the delegation credentials. The access control unit of Figure 9 can perform the verification for both primary signatures 24, 26 and the secondary signature 28 at the same time as is indicated by the broken lines. In the example of Figure 9, the secondary user is allowed by the ACLs of the delegation certificate 400a of primary owner of signature 1 to carry out operation [3] where the resource 100a, 100b allows according to its own ACL whilst the secondary user is allowed by the ACLs of the delegation certificate 400b of primary owner of signature 2 to carry out operations [2, 3] where the resource 100a, 100b allows.

In a further modification of the system, which may be combined with previous versions, the access control lists either attached to the resources or used in the delegation certificates can specify that any further delegation certificate must contain a "challenge" value. In this system when accessed by the code the access control unit 300 returns a "challenge" value which is unlikely to be guessed usually because the number is large and appears to be random. The access control unit 300 keeps a record of all the challenges that have been issued, along with information about the how long the challenge value is considered to be valid. For the code to be allowed access to operations permitted by an ACL entry requiring a challenge it must have a certificate that includes a currently valid challenge and this must be correctly signed. By controlling the lifetime of the challenge the system can control the lifetime of certificates used by the code. This can be used to grant temporary access rights to a body of code.

The differences between known systems and the system of the present invention will clearly be understood from the foregoing. In particular, in known systems, the code once authorised can access any available resource and the system requires the secure unit to protect the code, the resources and the checking process from hostile external attacks. In contrast, in the system of the present invention, the authorisation is contained in the resource access rather than the loading of the code and the secure sub-unit 180 not only protects the code, resources and the checking process from hostile external attacks but also protects the resources and checking processes from attacks from hostile code already loaded in the execution engine 200. This extra protection is necessary owing to the delayed nature of the authorisation of accesses to the resources.

It will be appreciated that the code, once loaded in the secure sub-unit 180, can perform any operation within the sub-unit including altering the code itself, since the authenticity of the code is checked before execution begins. It will also be appreciated that the present invention is more efficient than systems where the code is authenticated each time an access to the resources is made since the signature on the code need only be validated once.

In cases where more than one signature is applied to the code before it is loaded, all of the signatures are verified as the code is loaded and the identities of all the code signors are recorded. When an access to a resource is made, either the verification device can allow the access if any of the signors would have the right to make the access or, alternatively, the code might indicate in the request the identity whose credentials should be used for the authorisation.

In all of the systems represented in the drawings, various resources are required to be accessed by particular blocks of computer program code. The programs may be general purpose programs which make use of the various resources. An aim of the invention is to control the access to the resources so that only authorised programs may access the resources in authorised ways or cryptographic keys which may be accessed by selected users.

It is up to the owner of the computer system to determine which keys are to be trusted for which operations. Consequently, if further resources are added to the computer system, further keys must be added to the access control unit 300.

CLAIMS:

1 A method of controlling access to a resource (100) in a computer system by a body of code (22) having a signature (24) associated therewith, the method comprising the steps of:

(i) providing a cryptographic key (102) associated with said resource;

(ii) conducting a verification operation on said signature (24) using the cryptographic key (102) associated with said resource (100);

(iii) and controlling access to the resource (100) by the code (22) in dependence upon the result of said verification operation.

2 A method as claimed in claim 1 wherein:

said code (22) has a plurality of signatures (24, 28) associated therewith;

step (ii) comprises conducting a verification operation on each said signature (24, 28) using said cryptographic key (102) associated with said resource (100);

and step (iii) comprises allowing said code access to said resource in response to any one of said signatures being verified.

3 A method as claimed in claim 1 wherein:

said code (22) has a plurality of signatures (24, 28) associated therewith;

step (ii) comprises conducting a verification operation on each said signature (24, 28) using said cryptographic key (102) associated with said resource (100);

and step (iii) comprises allowing said code access to said resource only in response to all of said signatures being verified.

4 A method as claimed in any of claims 1 to 3 further comprising:

providing an execution engine (200) separate from said resource (100) for executing said code (22);

and step (iii) comprises loading said code into said execution engine in response to said signature being verified.

5 A method as claimed in any of claims 2 to 4 wherein:

said resource (100) has a plurality of associated keys (102);

step (ii) comprises conducting a verification operation on each said signature (24, 28) using each said cryptographic key (102) associated with said resource (100);

and step (iii) comprises allowing said code access to said resource in response to any one of said signatures being verified.

6 A method as claimed in any of claims 2 to 4 wherein:

said resource (100) has a plurality of associated keys (102);

step (ii) comprises conducting a verification operation on each said signature (24, 28) using each said cryptographic key (102) associated with said resource (100);

and step (iii) comprises allowing said code access to said resource only in response to all of said signatures being verified.

7 A method as claimed in claim 4 further comprising:

(iv) conducting a further verification operation on said signature (24) using the cryptographic key (102) associated with said resource (100) in response to said code (22) in said execution engine (200) attempting to access said resource;

(v) and further controlling access to the resource (100) by the code (22) in dependence upon the result of said verification operation.

8 A method as claimed in claim 7 wherein:

said code (22) has a plurality of signatures (24, 28) associated therewith;

step (iv) comprises conducting a verification operation on each said signature (24, 28) using said cryptographic key (102) associated with said resource (100);

and step (v) comprises allowing said code access to said resource in response to any one of said signatures being verified.

9 A method as claimed in claim 7 wherein:

said code (22) has a plurality of signatures (24, 28) associated therewith;

step (iv) comprises conducting a verification operation on each said signature (24, 28) using said cryptographic key (102) associated with said resource (100);

and step (v) comprises allowing said code access to said resource only in response to all of said signatures being verified.

10 A method as claimed in any of claims 7 to 9 wherein:

said resource (100) has a plurality of associated keys (102);

step (iv) comprises conducting a verification operation on each said signature (24, 28) using each said cryptographic key (102) associated with said resource (100);

and step (v) comprises allowing said code access to said resource in response to any one of said signatures being verified.

11 A method as claimed in any of claims 7 to 9 wherein:

said resource (100) has a plurality of associated keys (102);

step (iv) comprises conducting a verification operation on each said signature (24, 28) using each said cryptographic key (102) associated with said resource (100);

and step (v) comprises allowing said code access to said resource only in response to all of said signatures being verified.

12 A method as claimed in any of claims wherein the or each said key (102) is configured to authorise preselected actions on said resource (100).

13 A method as claimed in any of claims 1 to 11 wherein:

the or each said key (102) is configured to authorise preselected actions on said resource (100);

and following verification of the or one of the signatures by a said key associated with said resource, access to said resource by said code (22) is allowed only in relation to those preselected actions associated with said key.

14 A method as claimed in any of claims 1 to 13 comprising:

providing said code (22) with first and second signatures (24, 28);

associating a cryptographic key (402) for said second signature with said first signature;

step (ii) comprises conducting a verification operation on said first signature (24) using said cryptographic key (108) associated with said resource (100), and conducting a verification operation on said second signature (28) using said cryptographic key (402) associated with said first signature (24);

and step (iii) comprises allowing said code access to said resource in response to both of said signatures being verified.

15 A method as claimed in claim 14 wherein:

one of said keys (24, 108) is configured to authorise preselected actions on said resource (100);

and following verification of said signatures by said keys, access to said resource by said code (22) is allowed only in relation to those preselected actions associated with said one of said keys.

16 A method as claimed in claim 14 wherein:

each of said keys (24, 108) is configured to authorise preselected actions on said resource (100);

and following verification of said signatures by said keys, access to said resource by said code (22) is allowed only in relation to those preselected actions which are common to said keys.

17 A computer system comprising:

verification means (300) for verifying a digital signature associated with a body of program code (22);

and a resource having a cryptographic key associated therewith;

wherein said verification means is operable to use said cryptographic key to verify said digital signature thereby to allow access of the code to said resource in dependence upon the verification.

18 A system as claimed in claim 17 further comprising an execution engine means for running said program code;

wherein said execution engine means is separate from and is coupled to said resource by way of said verification means (300);

and said verification means is operable to verify said code both prior to said code loading into said execution engine means and prior to said code accessing said resource from said execution engine means.

19 A system as claimed in claim 18 wherein said verification means, said resource and said execution engine means are contained within a secure module.

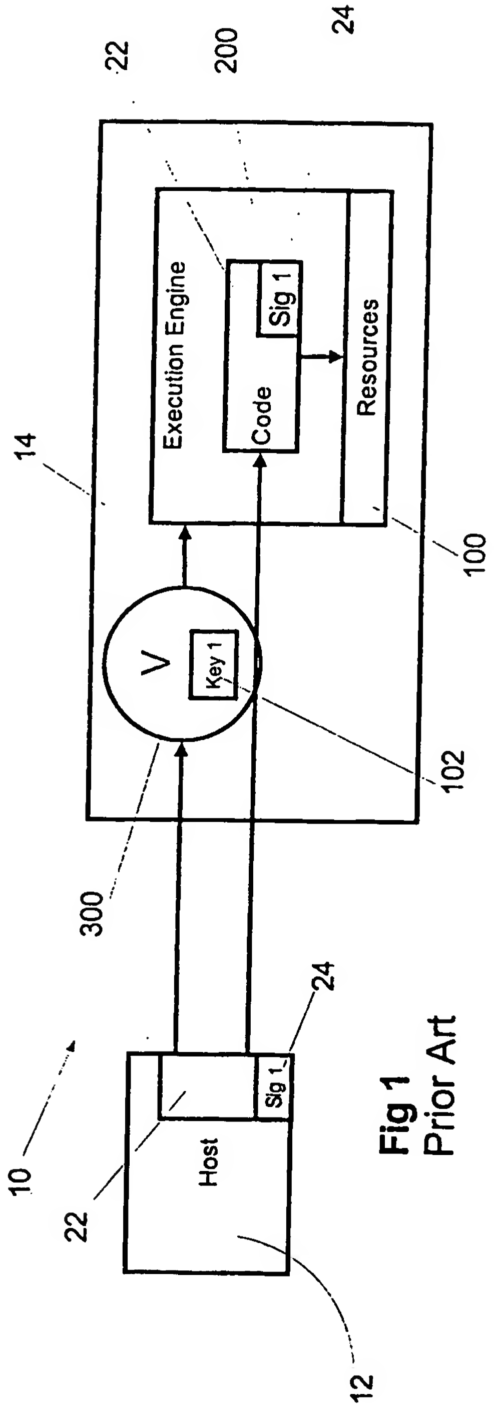


Fig 1
Prior Art

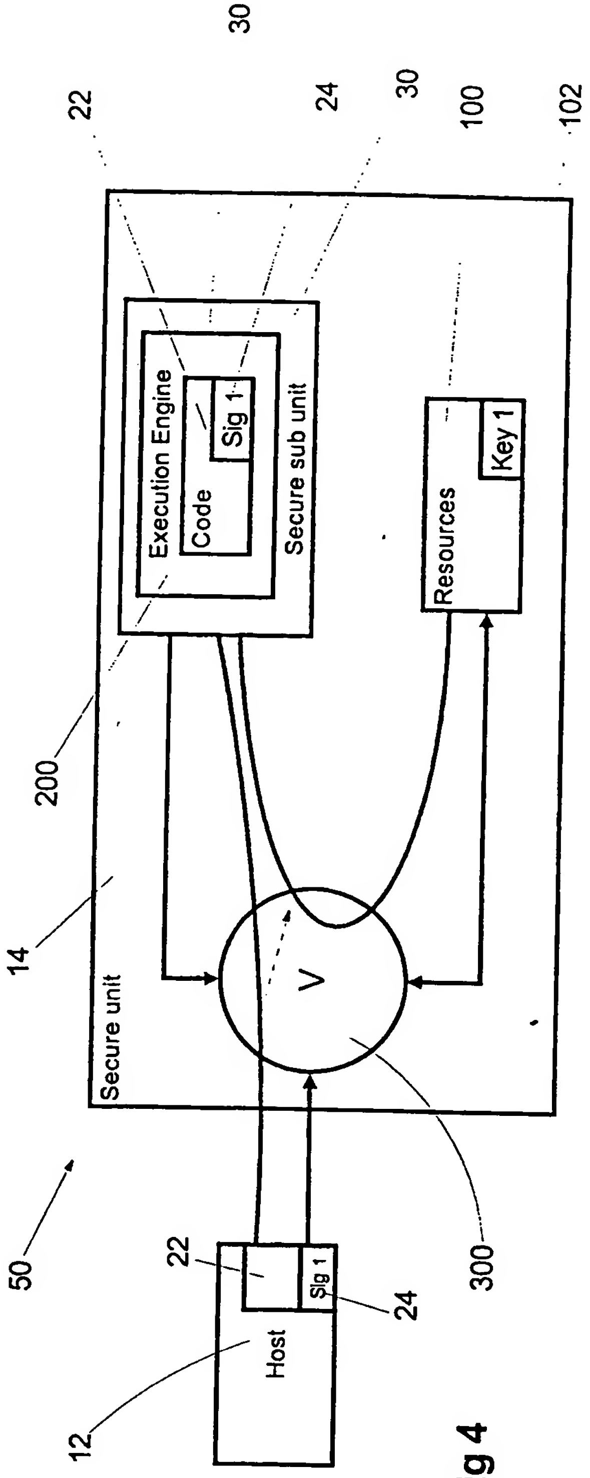


Fig 4

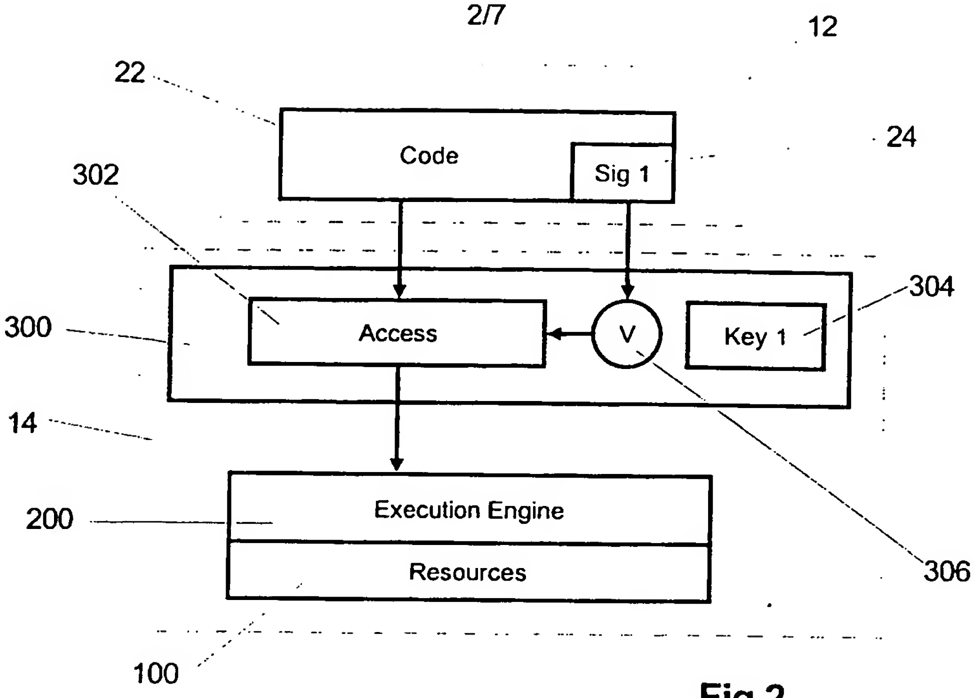


Fig 2
Prior Art

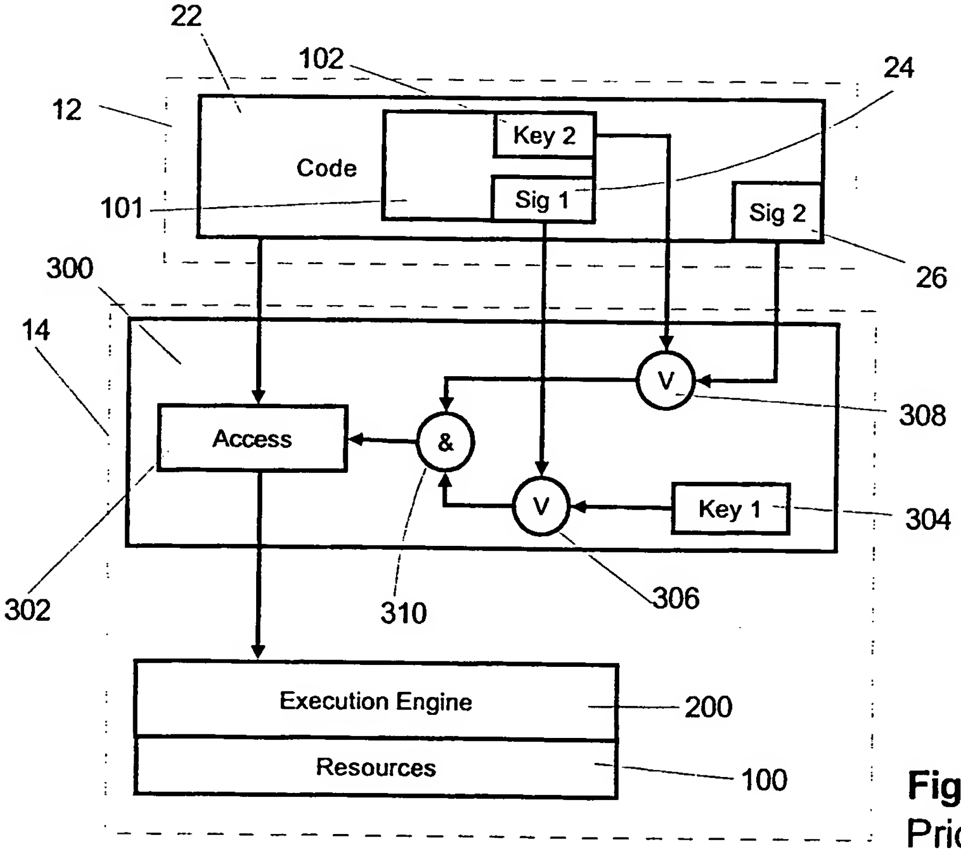


Fig 3
Prior Art

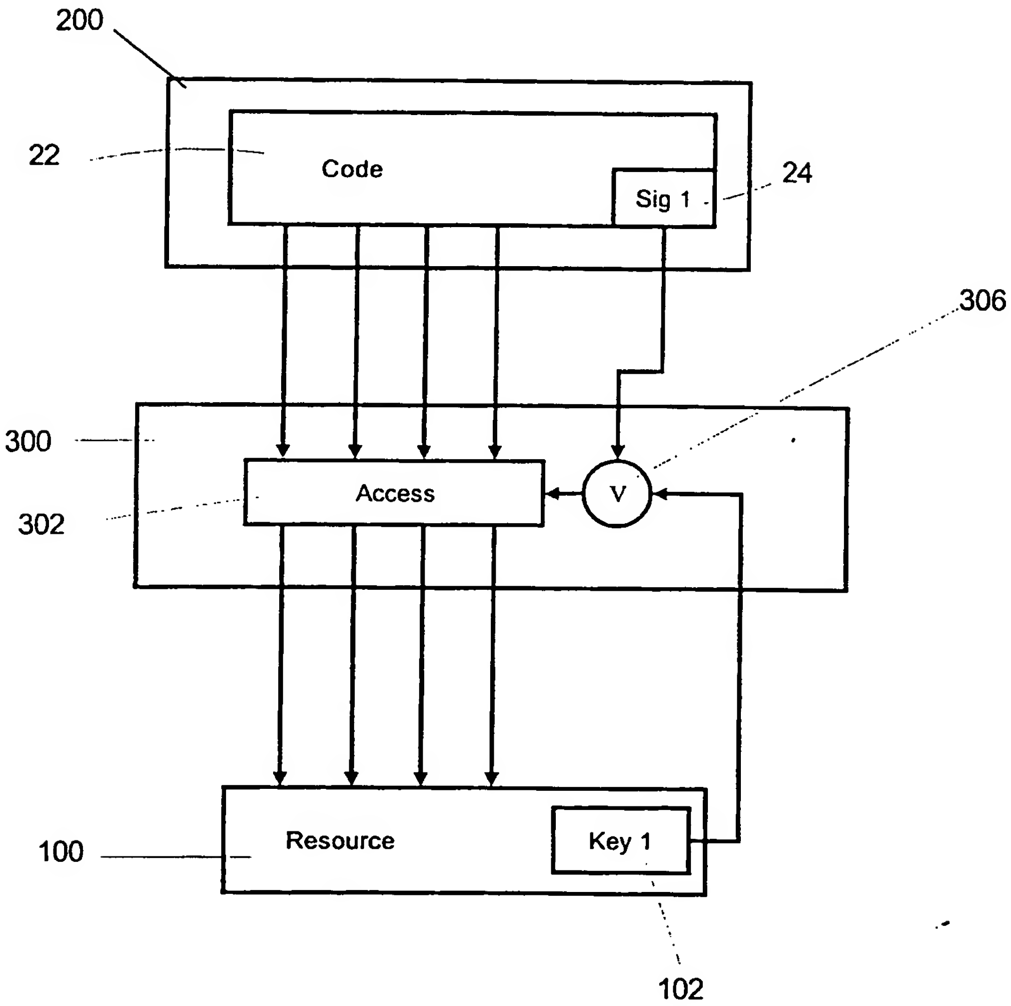


Fig 5

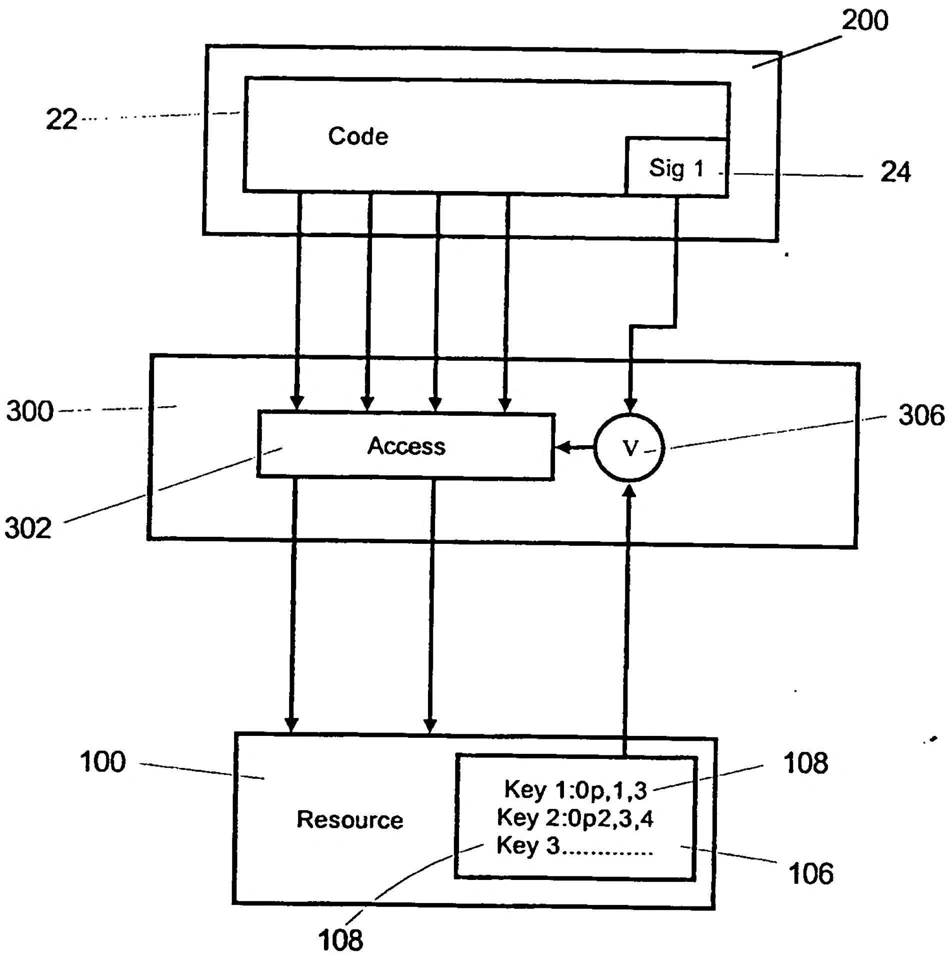

200
22
Code
Sig 1
24
300
Access
V
306
302
100
Resource
Key 1:0p,1,3
Key 2:0p2,3,4
Key 3.............
108
106
108

Fig 6

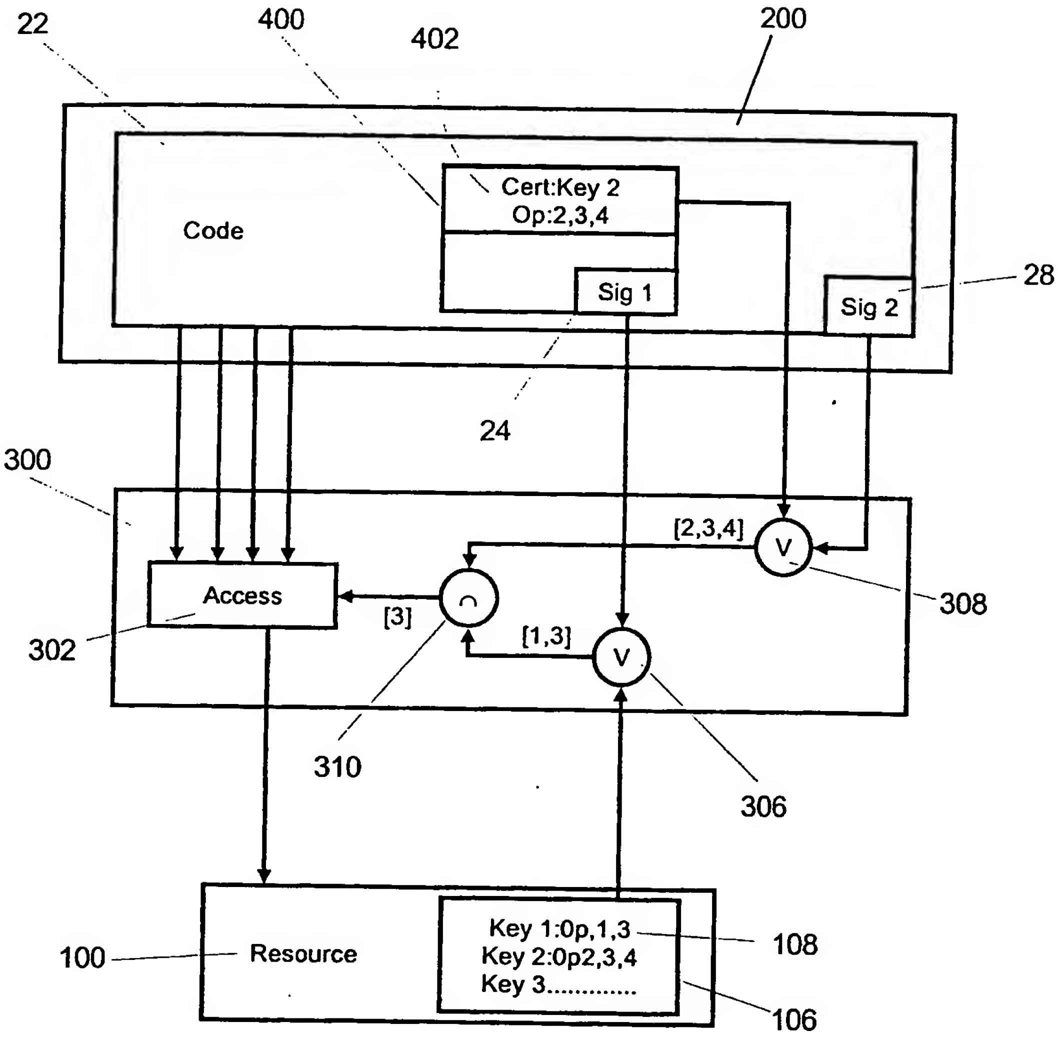

22
400
402
200
Cert:Key 2
Op:2,3,4
Code
Sig 1
Sig 2
28
24
300
[2,3,4]
V
Access
∩
[3]
[1,3]
V
308
302
310
306
100
Resource
Key 1:0p,1,3
Key 2:0p2,3,4
Key 3............
108
106

Fig 7

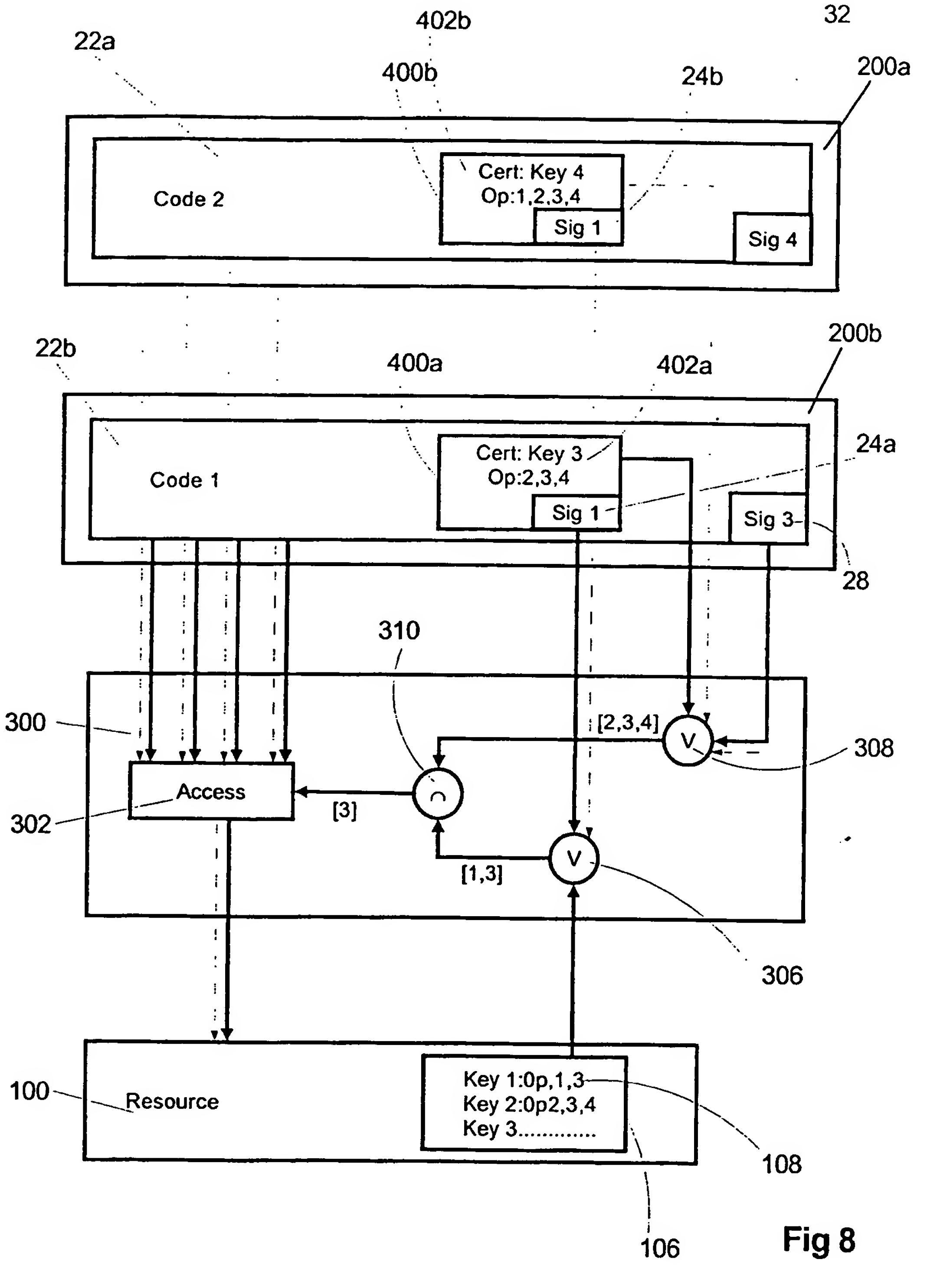

32
22a
402b
400b
24b
200a
Code 2
Cert: Key 4
Op:1,2,3,4
Sig 1
Sig 4
22b
400a
402a
200b
24a
Code 1
Cert: Key 3
Op:2,3,4
Sig 1
Sig 3
28
310
300
[2,3,4]
308
V
Access
302
[3]
∩
V
[1,3]
306
100
Resource
Key 1:0p,1,3
Key 2:0p2,3,4
Key 3............
108
106

Fig 8

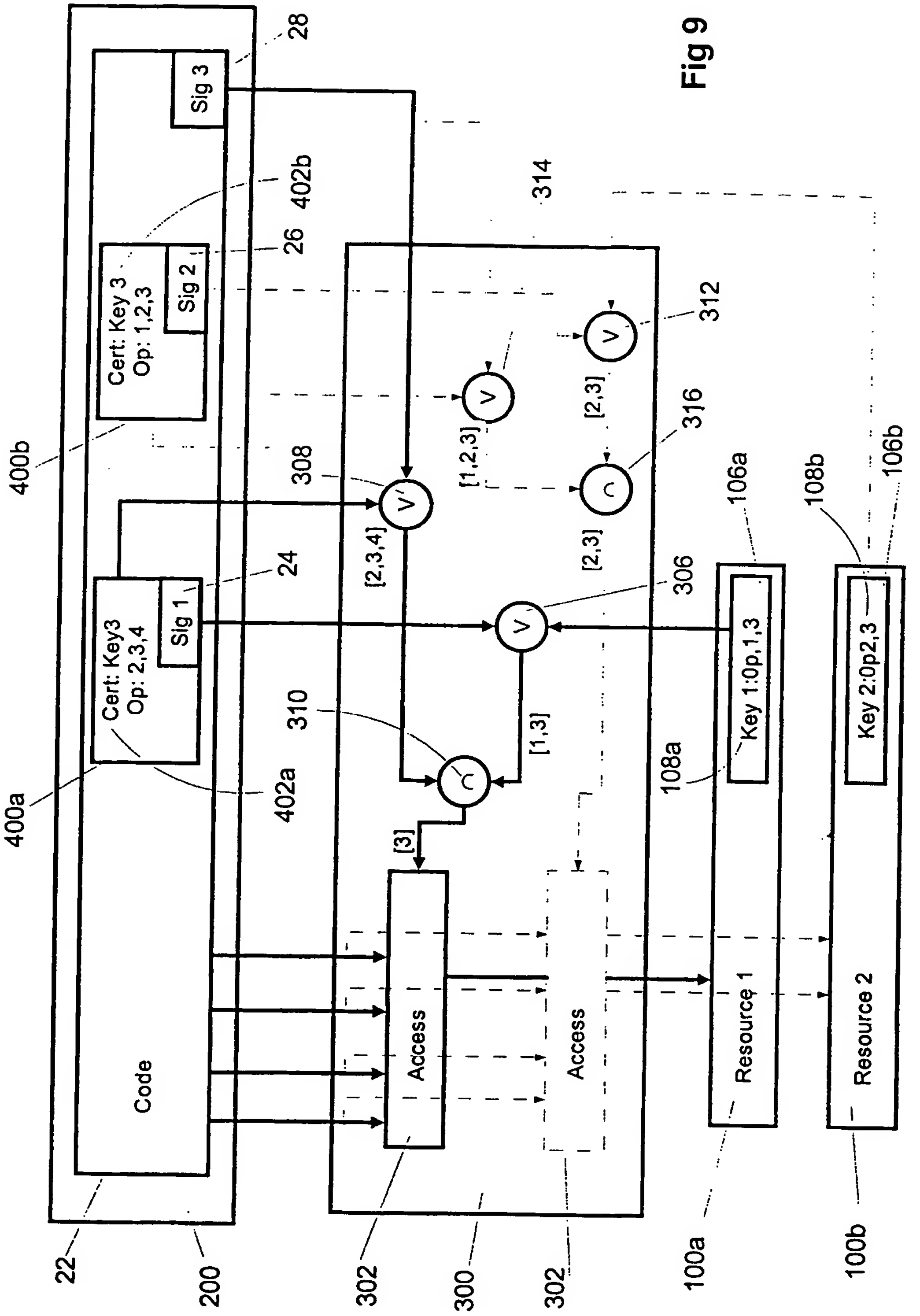

Sig 3
28
402b
26
Cert: Key 3
Op: 1,2,3
Sig 2
400b
314
312
V
V
[1,2,3]
[2,3]
316
C
[2,3]
308
V'
[2,3,4]
24
Cert: Key3
Op: 2,3,4
Sig 1
306
V
106a
108b
106b
Key 1:0p,1,3
Key 2:0p2,3
310
C
[1,3]
400a
402a
108a
[3]
Access
Access
Code
Resource 1
Resource 2
22
200
302
300
302
100a
100b

Fig 9

INTERNATIONAL SEARCH REPORT

In ational Application No
PCT/GB 01/00688

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

PAJ, EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No
X	EP 0 969 366 A (SUN MICROSYSTEMS INC) 5 January 2000 (2000-01-05) abstract	1,3,6, 12-17
Y	page 4, column 45 -page 8, column 45 figures 3,4 ---	2,4,5,7, 9,11,18

-/--

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 7 June 2001	Date of mailing of the international search report 15/06/2001
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Arbutina, L

1

Ir ational Application No
PCT/GB 01/00688

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication where appropriate, of the relevant passages	Relevant to claim No
Y	GONG L ET AL: "Going beyond the sandbox: an overview of the new security architecture in the JavaDevelopment Kit 1.2" PROCEEDINGS OF THE USENIX SYMPOSIUM ON INTERNET TECHNOLOGIES AND SYSTEMS, 8 December 1997 (1997-12-08), XP002100907 page 103, right-hand column, paragraph 1.1 page 104, left-hand column, paragraph 1.2 page 105, right-hand column, paragraph 2.1 page 108, right-hand column, paragraph 2.5 figure 3 ---	4.7.9. 11.18
Y	US 5 337 360 A (FISCHER ADDISON M) 9 August 1994 (1994-08-09)	2,5
A	abstract ---	8,10
A	WO 98 19237 A (SCHLUMBERGER TECHNOLOGIES INC) 7 May 1998 (1998-05-07) abstract figures 1,12 -----	19

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

Ir ational Application No
PCT/GB 01/00688

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0969366 A	05-01-2000	US 6138235 A	24-10-2000
		JP 2000148469 A	30-05-2000
US 5337360 A	09-08-1994	AT 198800 T	15-02-2001
		AU 3560793 A	07-10-1993
		CA 2093094 A	07-10-1993
		DE 69329869 D	22-02-2001
		DK 565314 T	19-03-2001
		EP 1031908 A	30-08-2000
		EP 0565314 A	13-10-1993
		ES 2153371 T	01-03-2001
		JP 6295286 A	21-10-1994
		US 5390247 A	14-02-1995
WO 9819237 A	07-05-1998	AU 722463 B	03-08-2000
		AU 4911897 A	22-05-1998
		EP 0932865 A	04-08-1999
		JP 2000514584 T	31-10-2000